STRUCTURAL HEALTH MONITORING WITH LARGE DATA SETS

by

Spencer Shiveley

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

The University of Utah

May 2017

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Spencer Shiveley**
has been approved by the following supervisory committee members:

Joel Harley ,	Chair(s)	**09 Mar 2017** Date Approved
Behrouz Farhang ,	Member	**09 Mar 2017** Date Approved
Neal Patwari ,	Member	**09 Mar 2017** Date Approved

by **Gianluca Lazzi** , Chair/Dean of
the Department/College/School of **Electrical and Computer Engineering**
and by **David B. Kieda** , Dean of The Graduate School.

ABSTRACT

Structural health monitoring systems collect and process large volumes of data taken over many years of a structure's service. Ultrasonic guided wave systems, in particular, must process an abundance of time-domain waveform data from widely distributed sensors. As few as 8 sensors that transmit and receive ultrasonic waves in pitch-catch mode every 10 minutes can accumulate over one terabyte of data in five to ten years. This number quickly rises as systems grow in size and complexity. As a result, computation and storage efficiency is extremely important, and current guided wave damage detection technologies cannot efficiently process such large data sets. This thesis starts with an introduction and survey of the structural health monitoring and data compression fields. A dimensionality reduction technique using random projections is proposed. The potential for dimensionality reduction method for improving computation time and storage efficiency is discussed. Random projections using sparse matrices is investigated as a tool in implementing a real-time structural health monitoring system with singular value decomposition as a damage detection method. At the end, future directions for research to make this technology more viable in application are suggested.

For my wife, Kacey, and unborn child.

CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor Dr. Joel Harley for his continual support and guidance throughout my graduate studies and thesis research. Without him, this research would not be possible. I am grateful for all of the opportunities he has provided me over the past year and for igniting a passion in the field of signal processing. It was a privilege and honor to be one of his first graduate students and I thank him for all the effort and time he has put into helping me develop my skills.

A special thanks to my family. Words can not express how grateful I am to my mother and father for all of the sacrifices that you have made on my behalf. Your encouragement for me was what sustained me thus far.

I would especially like to thank my beautiful wife, Kacey. Thank you for supporting me during this difficult time, and for putting up with late nights and days without seeing each other. Without you, none of this would be possible.

CHAPTER 1

INTRODUCTION

1.1 Motivation

According to the U.S. Department of Transportation, pipeline incidents have killed over 300 people, injured an additional 1,300, and caused over 7 billion dollars in damage over the past 20 years [1]. Furthermore, bridge inspections during the late 1980s showed that of the 576,000 US highway bridges, 236,000 were rated deficient by present-day standards [2]. As a result of these disasters, structural health monitoring (SHM) systems are being researched to increase safety and reduce costs by preventing catastrophic failures in our structure.

The main component of an SHM system is nondestructive evaluation. Nondestructive evaluation is the practice of detecting, classifying, and locating damage in materials and structures. SHM is the process of performing and monitoring these evaluations over time, often with *in situ* sensor networks.

There are two main classes of SHM systems: passive and active. Passive SHM use sensors to "listen" over time for strain/loading on the structure, environmental conditions, and acoustic emissions due to cracks [3]. This data is then compared to existing models. Active SHM, on the other hand, uses sensors to "interrogate" the structure by transmitting a signal from a single sensor while the remaining sensors listen for this transmission. The data is then fed back into the system where signal processing techniques are used to determine the state of the structure [4].

The use of guided waves is a common approach for active SHM due to their ability to propagate long distances [5]. Guided waves are defined as stress waves that follow a path defined by the geometry of the structures [3]. When guided waves are incident on structural discontinuities (such as boundaries or damage), the guided wave will scatter in all directions. Owing to this complex nature of guided waves, knowledge of the structure is needed *a priori* to discriminate scat-

tered waves caused by the structure's boundaries from those caused by damage. This knowledge comes from complex models or baseline measurements of the structure in a known healthy state [3]. Current measurements are then compared to the baseline to determine the state of the structure.

Singular value decomposition [6] is one damage detection method that has been recently introduced to extract damage variations from entire data sets. This strategy exploits all of the available data to more effectively extract critical trends and achieve robust damage detection. Yet, as SHM data sets grow over years of operation, the efficient storage of this data and computational feasibility of the these types of algorithms will become a significant issue. As a result, there is a growing need to improve the efficiency of these methods.

This thesis addresses this challenge by integrating singular value decomposition damage detection with random projection theory [7]. Random projection theory has been used extensively in big data analysis [8]. It allows for the compression of data while approximately retaining linear similarity metrics (such as correlation). As a result, we can accurately detect damage with high storage and computational efficiency.

1.2 The Field of Structural Health Monitoring

Structural Health Monitoring is an emerging technology that seeks to give a real-time diagnosis of the "damage state" of the structure under observation. SHM systems are developed for the following tasks: detect damage, classify damage, determine damage extent, locate damage, and give a prognosis of the remaining life of the structure [9]. Due to their ability to complete these tasks, SHM systems are of particular interest in aviation, oil and gas, and construction industries. In the following sections, we give a discussion on the motivations associated with SHM, a brief history of SHM, the tasks SHM seeks to accomplish, and recent work in the field of SHM.

1.2.1 Motivations for SHM

Complex structures are typically created with an anticipated lifetime and are retired after a set number of years. Aircrafts, for example, are one type of complex structure that are aging in the United States and around the world. The average aircraft, in the United States, is over 15 years old and the airline industry spends over 6 billion dollars annually in maintenance and inspections for their fleets [10]. Airlines utilize schedule-based maintenance programs that involve four levels of "checks", each increasing in detail. "A-checks" are the most frequent and require the least amount of detail. These typically occur after 500 flight hours and require approximately 150 man hours. "B-checks" and "C-checks" occur less frequently but require a more thorough inspection and the aircraft is out of commission for longer periods. "D-checks" are the least common, occurring 4-6 times throughout the service life of the aircraft. "D-checks" require major disassembling and can take tens of thousands of man hours, putting the aircraft out of service for several weeks [11].

Advances in SHM systems will allow current schedule-based inspections to evolve into more cost-efficient condition-based maintenance schedules [2]. Condition-based maintenance is a program that allows for maintenance decisions based on information collected from SHM systems [12]. This will avoid catastrophic failures, minimize costly downtime, and allow optimal use of the structure to operate for a maximized lifetime.

In addition to the economical benefits, safety improvements are strong motivational factors. The following two accidents are examples of unsatisfactory maintenance. Aloha Airlines flight 243 was an accident in 1988 where the aircraft suffered extensive damage after an explosive decompression in flight, killing one and injuring 65 others. Investigations determined that sufficient inspection and maintenance would have avoided the failure [13]. In addition, the collapse of the Mianus River bridge in 1983 killed several motorists and injured several others. Investigations report the structure was not properly inspected and corrosion of key parts led to the collapse [14]. Proper implementation of SHM systems can greatly improve the safety of new and existing structures as well as reduce the operational

costs.

1.2.2 Tasks of SHM

SHM is a complex system comprised of multiple subsystems such as: sensor integration, data transmission, computational power, processing of data, and storage of data [2]. The processing of data subsystem for damage detection can be described in a five-step process [9, 15].

- *Existence* - Is the structure damaged?
- *Location* - Where is the structure damaged?
- *Type* - What type of damage does the structure contain?
- *Extent* - How bad is the structure damaged?
- *Prognosis* - What is the remaining lifespan of the structure?

Answers to these questions are a crucial step in a SHM system. Each answer will provide engineers with more knowledge about the structure and what course of action to take. Unfortunately, each question is increasingly difficult to answer, requiring more baseline information and a more complex system [15, 16].

1.2.3 Brief History and Background: Damage Detection

Structural health monitoring has existed in one form or another throughout the years. Tap tests were common in the railroad and oil and gas industries. Technicians would move down the railroad or pipeline, occasionally tapping and listening for abnormalities [15]. More recently, logs made by mechanics and pilots during scheduled inspections are a common practice in aviation [11].

With the advent of the computer, more sophisticated methods for damage detection began to be explored. During the late 1970s and early 1980s, a lot of work was done in the oil and gas industry as well as at NASA for vibration-based SHM. Simulations for common damage scenarios were performed. Any changes in resonant frequencies were then correlated with empirical measurements of the structure [15]. Further advances in electronics, memory storage, and signal processing

has led to the modern field of SHM [17]. Due to these technological advancements, guided wave-based SHM has received a lot of attention in the past two decades.

1.2.4 Ultrasonic Guided Waves

Ultrasonic guided waves are a popular tool in active SHM schemes. There are several reasons guided waves are attractive: (1) due to the advancement in electronics, sensors and transducers have become exceedingly small and cheap; (2) they are simple to implement and generate, (3) are sensitive to many types of damage [5], and (4) can propagate long distances with little attenuation and therefore, can cover large areas with relatively few sensors [18]. Guided waves are also an attractive tool because they are already a well-established practice in the nondestructive testing industry [5]. Therefore, many signal processing methods have been developed for effective damage detection, classification, and localization.

Guided waves are mechanical stress waves that propagate along a path determined by the boundaries of the structure [3]. For example, acoustic waves guided by the geometry of long, wide plates with a finite thickness are a type of guided waves. The boundaries of the plate reflect the acoustic waves back and forth, guiding them down the direction of the plate's length and width [19]. One specific class of guided waves that is dealt with in this thesis is Lamb waves.

1.2.4.1 Lamb Waves

Lamb waves are elastic longitudinal waves that exist in thin, infinite plates and were first theorized by an English mathematician Horace Lamb in 1917 [9]. More extensive theoretical frameworks were developed by others in the following decades. In the early 1960s, Worlton experimented with Lamb waves as a form of damage detection [20]. Due to the abundance of many plate-like structures, e.g., an airplane's wing, Lamb waves are a popular class of guided waves in SHM [3].

Another attractive reason for Lamb wave-based SHM is the many advantages they possess: (1) transducers are cheap and light-weight and can easily be incorporated into the material during construction of a structure; (2) due to the fact that they are multimodal in nature, Lamb waves are able to identify multiple defects;

(3) higher frequency content in transmitted signals lend to being able to detect small, even millimeter length, damage; (4) due to the structure's guidance of the wave, there is little attenuation and large areas can be interrogated with relatively few sensors; (5) expensive and complicated equipment for rotating and vibrating the structure is unnecessary; (6) guided wave transducers require little energy consumption making them an ideal candidate for long-term use for inaccessible structures [5, 9].

The numerous advantages of Lamb wave-based SHM do come with some trade-offs. Due to the fact that the waves consist of multiple wave-modes, all propagating simultaneously, a received Lamb wave signal is quite complex. Furthermore, the high velocity of the waves can cause multipath reflections from the structures boundaries to mask any reflections due to damage [21]. Lamb waves are also highly dispersive; the shape of the wave changes as it propagates, i.e., group and phase velocities are dependent on frequency [4]. Finally, Lamb waves are sensitive to high-frequency ambient noise, low-frequency structural vibrations, and environmental and operating conditions such as temperature [17].

One commonly used approach for Lamb wave-based SHM is a pitch-catch technique [3]. In this method, a pulse signal is sent across a structure where a sensor, located elsewhere, will receive the signal. Based on the time-of-flight, amplitude, frequency, and phase of the received signal, abnormalities in the structure can be identified [21]. Lamb wave-based SHM relies heavily on signal processing methods to detect and classify damage. Current research in guided wave SHM is largely focused on signal processing techniques to extract information from the signal charactersitics and provide a solution for any or all of the tasks listed in section 1.2.2. In the following section, we go into some detail on two popular signal processing techniques used in guided wave SHM.

1.2.5 Recent Work

1.2.5.1 Baseline Subtraction

Guided waves are inherently complex signals. In an undamaged structure, the received signal will consist of many interfering reflections from the surfaces and

boundaries of the structure. These interfering reflections are detrimental because they can mask damage signals [22]. One simple method of damage detection is to compare snapshots of the structure before and after damage occurs. The simplest method is to subtract the two signals [23]. When damage occurs, the guided waves will scatter from the damage, creating new interfering reflections. However, existing reflections from artifacts and boundaries will remain unchanged. As a result, upon subtraction, reflections due to benign structural features will be removed, leaving only those pertaining to damage [23–25].

While this approach works well in ideal circumstances, it is well known that environmental and operating conditions, such as low-frequency vibrations, changes in humidity, and temperature effects, lead to fundamental changes in the structure's properties, such as expansion and contraction of the materials [26]. This change in the physical properties leads to changes in the propagating wave that can be misinterpreted as damage when compared with a baseline [23].

Temperature change is the environmental condition that has the largest detrimental effect on guided waves. Two main bodies of research have formed in order to compensate for temperature change: restoring the amplitude and phase the guided wave had when the baseline signal was taken [26–28] and comparing with look-up tables of signals representing environmental conditions that the structure would commonly see [18, 25]. These temperature compensation methods are typically referred to in literature as baseline signal stretch (BSS) and optimal baseline selection (OBS), respectively [22]. However each method has its limitations. BSS methods model temperature changes as having a stretching or compressing effect on the signal. This is only an approximate model. BSS also assumes temperature affects the structure uniformly, which is not necessarily true. OBS, on the other hand, requires multiple baseline signals resulting in a sizable amount of data [26]. Furthermore, this database has the baseline signals taken at discrete temperatures, when in reality, the structure will be subject to a continuous range of temperatures. These baselines can be difficult to obtain and introduce errors due to the discretization of temperature effects [6].

1.2.5.2 Singular Value Decomposition

Singular value decomposition (SVD) has recently been introduced as a new temperature compensation and damage detection framework [6]. SVD extracts critical trends across the data set and achieves robust damage detection. SVD is a linear decomposition method that is closely related to principle component analysis and commonly used in dimensionality reduction. SVD is beneficial because it does not require a baseline or depend on models detailing temperature affect on guided waves, and it has the ability to reduce noise by looking across multiple measurements [29]. As the name suggests, this strategy is implemented by computing the SVD of the data set. Time-domain measurements from a single sensor are gathered into a $Q \times M$ data matrix. In this data matrix, each column of length Q represents a measurement in "fast time", a single measurement taken over several milliseconds. Each row of length M represents a single instance of the guided wave signal over days, weeks, months, and years [6, 29, 30]. SVD then decomposes the data into a matrix product

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^H \tag{1.1}$$

In (1.1), the columns of $\mathbf{U}$ contain the left singular vectors of the data matrix, $\mathbf{X}$, $\mathbf{S}$ is a diagonal matrix where the elements are the singular values of $\mathbf{X}$, and the columns of $\mathbf{V}$ contain the right singular vectors of $\mathbf{X}$. SVD will decompose the principle variations in slow time as the left singular vectors [6, 29] . As a result, the columns of $\mathbf{V}$ will represent "slow time" trends such as slow variations in temperature. If damage were to occur, one of the left singular vectors would show the corresponding change in "fast time" [29]. Thus, in addition to being a robust temperature compensation and damage detection framework, SVD-based damage detection can also perform a robust baseline subtraction.

1.3 Data Compression

Since the development of the transistor, the world has seen advances in electronics and digital hardware on an exponential scale. This has started a digital revolution and brought the world into what is commonly referred to as the "Information Age." Due to this digital revolution, researchers in many fields have

begun collecting enormous amounts of data to analyze. During that past few decades, the amount of digital information stored has nearly doubled every 2.5 years. By the year 2007, approximately 2 zettabytes of optimally compressed data were digitally stored [31]. This trend is expected to continue if not increase with the ever-growing Internet. Although significant advances have been made in data storage, it is simply not enough. The need to store and transmit data is steadily outpacing the development of mass storage [32]. Data compression, the ability to represent information in a compact form, is the only solution. This is done by identifying and exploiting the structure of the data [32].

Structural health monitoring is not exempt from this issue. When SHM systems are implemented, sensors will begin collecting data the moment the system is turned on. As a result, data storage is a concern that needs to be addressed. Data collected throughout the service life of the structure needs to be efficiently compressed for storage. However, this data will also be periodically recalled and analyzed and therefore, much of the information contained in the compressed data needs to be maintained.

Taking a data matrix $\mathbf{X}$ and forming a compact representation $\mathbf{X_c}$ is only the first half of data compression. In order to obtain an estimate of the original data $\hat{\mathbf{X}}$, a reconstruction step is needed. There are two main classes of data compression: lossless compression and lossy compression. In general, lossy compression can provide much higher compression rates than lossless compression but at the expense that $\hat{X}$ will differ from X [32].

In this thesis, we focus on lossy compression, and in particular, dimensionality reduction. Dimensionality reduction is defined as finding a low-dimensional manifold that embeds the high-dimensional data [33]. The following two sections give a brief introduction into two popular techniques in dimensionality reduction: principal component analysis and random projections.

1.3.1 Principal Component Analysis

Principal component analysis (PCA) is a statistical method that takes a set of vectors, with possibly correlated directions, and maps them to a set of vectors

with orthogonal directions (principal directions/components) via an orthogonal transformation. The principal components form an orthogonal basis in which the first principal component maximizes the variance of the data set. The second principal component will then maximize the variance of the data set subject to the constraint that it must be orthogonal to the first principal component and so on. Commonly, the first k principal components can describe most of the variance in a data set [34]. The remaining principal components can then be discarded with little loss in information. PCA is regarded as the optimal linear decomposition in the mean-squared sense [35]. PCA is widely used in many fields and is very similar to singular value decomposition and the Karhunen-Loéve transformation [36].

PCA is a linear decomposition based on the covariance matrix [34] and has several interpretations depending on the field of study. In a statistical framework, PCA can be thought of as method to decorrelate related variables, selecting those that best represent the entire data set [37]. PCA is also a spectral decomposition method as it projects the data along the eigenvectors of the correlation matrix.

1.3.1.1 PCA Relation to Singular Value Decomposition

Suppose we have the $Q \times M$ standardized data matrix $\mathbf{X}$, detailed in section 1.2.5.2. The empirical covairance matrix for the guided wave samples is given by $\mathbf{C} = \frac{1}{M-1}\mathbf{X}\mathbf{X}^T$. This $\mathbf{C}$ matrix is a symmetric semipositive definite matrix and can therefore be diagonalized as

$$\begin{aligned} \mathbf{C} &= \frac{1}{M-1}\mathbf{X}\mathbf{X}^T \\ &= \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \end{aligned} \tag{1.2}$$

where $\mathbf{U}$ is a matrix of eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues in descending order, i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_Q \geq 0$. As mentioned previously, PCA is a spectral decomposition where the eigenvectors are called *principle directions* [38]. The projection of the data matrix onto the principal directions are the *principal components*, where the i-th principal component is given by the i-th column of $\mathbf{X^T U}$. As was shown in section 1.2.5.2, the singular value decomposition of the

data matrix, $\mathbf{X}$, is given by

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{1.3}$$

substituting (1.3) into (1.2), one can easily see that

$$\begin{aligned} \mathbf{C} &= \frac{1}{M-1}\mathbf{X}\mathbf{X}^T \\ &= \frac{1}{M-1}(\mathbf{U}\mathbf{S}\mathbf{V}^T)(\mathbf{U}\mathbf{S}\mathbf{V}^T)^T \\ &= \frac{1}{M-1}\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T \\ &= \mathbf{U}\frac{\mathbf{S}^2}{M-1}\mathbf{U}^T \end{aligned} \tag{1.4}$$

Upon closer inspection, a few things are obvious from (1.4). First, the principle directions are given by the left singular vectors in $\mathbf{U}$. Second, the eigenvalues of $\mathbf{C}$ are related to the singular values of $\mathbf{X}$ by $\mathbf{\Lambda} = \frac{\mathbf{S^2}}{M-1}$. Thirdly, the *principal components* of $\mathbf{X}$ are given by the columns of $\mathbf{VS}$, $\mathbf{X^TU} = (\mathbf{U}\mathbf{S}\mathbf{V}^T)^T\mathbf{U} = \mathbf{VS}$.

1.3.1.2 PCA Applied to Structural Health Monitoring

Although principal component analysis is one of the most widely used linear dimensionality reduction methods, it is ill-suited for structural health monitoring for two reasons. Firstly, computation-time for PCA, much like SVD, is directly proportional to the size of the data matrix. Structural health monitoring data sets will grow in size as time goes on, making PCA computationally infeasible. Secondly, the key concept behind PCA is the span of the k-dimensional basis, formed by the first k eigenvectors of the covariance matrix $\mathbf{C}$. The span of this basis has the smallest deviation from the original data matrix $\mathbf{X}$.[34]. As a result, in PCA dimensionality reduction, only the k most significant eigenvectors are kept while the remaining $Q - k$ eigenvectors are discarded. However, in section 1.2.5.2, we showed that one of the left singular vectors, which is one of the principal components of $\mathbf{X}\mathbf{X}^T$, will correspond to damage. Depending on the severity of damage, this singular vector may or may not be among the k most significant singular vectors, and hence may not be one of k principal components. As a result, PCA dimensionality reduction may throw away important information regard-

ing damage. Therefore, we consider another approach for dimensional reduction based on random projections.

1.3.2 Low-dimensional Subspace Embeddings

An alternative approach is to find a mapping that takes data in $\mathbb{R}^Q \rightarrow \mathbb{R}^k$ where $k \ll Q$. Let us begin by considering a $Q \times M$ matrix $\mathbf{X}$; however, in this setting, it will be useful to consider this as a set of M points in a Q dimensional space. The Johnson-Lindenstrauss lemma states that a set of points in a high-dimensional space can be embedded into a lower dimensional space where the pair-wise (euclidean) distances, between points, can be approximately preserved with high probability. Interestingly enough, the Johnson-Lindenstrauss lemma is a by-product of a proof that extends Lipschitz mappings to Hilbert spaces; the authors needed the lemma in order to complete the proof [39]. Since then, this lemma has seen widespread application in computer science due to its implication of compression [40].

Advances in data collection and storage capabilities have allowed researchers to collect enormous amounts of data. However, many analysis techniques are slow and bottlenecked due to a large number of features or dimensions. This is often referred to as the "curse of dimensionality." Many existing data analysis tools are exponential in time with regards to dimensionality and therefore scale poorly with size. The JL lemma is one viable solution to this challenge. Due to the JL lemma's ability to reduce dimensionality, while maintaining euclidean distances between features, many existing analysis tools can be applied to big data sets with little loss in accuracy. Below is a formal definition of the JL lemma.

LEMMA 1 *[39]. Given a tolerance* $0 < \epsilon < 1$, *and an integer* M, *let* k *be a positive integer where* $k \geq (\epsilon^2 - \frac{\epsilon^3}{3})^{-1} \log M$. *Then for any set* $\mathbf{V}$ *of* M *points in* $\mathbb{R}^Q$, *there exists a mapping* $f : \mathbb{R}^Q \rightarrow \mathbb{R}^k$ *such that for* ***all*** $\mathbf{u}, \mathbf{v} \in \mathbf{V}$,

$$(1-\epsilon)||\mathbf{u}-\mathbf{v}||^2 \leq ||f(\mathbf{u}) - f(\mathbf{v})||^2 \leq (1+\epsilon)||\mathbf{u}-\mathbf{v}||^2. \tag{1.5}$$

There are a couple of important properties to notice about the JL lemma. First, (1.5) shows that the lower dimensional subspace $\mathbb{R}^k$ is inversely proportional to the "tolerance" ϵ and therefore implies that the mapping is good for reduction from a

high space to a "medium" space. In other words, if we require a strict tolerance for distortion, we must project into a sufficiently high dimension k. Therefore, if we wanted to perfectly preserve all pairwise distances, we cannot reduce the dimensions. Second, the cardinality of the subspace does not depend on the dimension Q of the higher space but rather the number of points M we wish to embed into the subspace. This means that the dimension into which we wish to embed our points only depends on the number of pairwise distances we wish to preserve (the distance between any two columns in $\mathbf{X}$). The more data we collect, the higher k must be. However, this scales logarithmically with measurements instead of polynomially or exponentially.

This lemma finds use in many computer science fields such as nearest neighbor search [41], clustering [42], compressed sensing [43], and various other machine learning algorithms. Later we will show that this lemma can be used in singular value decomposition, which we will apply to structural health monitoring data sets for damage detection.

1.3.2.1 Finding the Random Projection

The Johnson-Lindenstrauss lemma would not be very useful if it only told us that there exists a mapping $f : \mathbb{R}^Q \to \mathbb{R}^k$. However, the proof for the lemma in [39] relies heavily on the isoperimetric inequality property for a volume defined in a Q dimensional space [44]. Isoperimetric problems ask for sets or objects whose boundary is smallest for a given volume. In our case, we have a notion of volume defined by M points in $\mathbb{R}^Q$ and we wish to find the smallest (possibly scaled) boundary for this scenario. This is in fact an old problem dating back to ancient Greece, to determine the shape in the plane for which the perimeter is minimized subject to a volume constraint. The answer is, in fact, a circle and was known in ancient Greece; however, a rigorous proof did not exist until the 19th century. This shape turns into a sphere in higher dimensions and therefore, in order to find the smallest shape that "preserves" the volume of our data, we must project onto the surface of a sphere. So how exactly do we choose the mapping f? We choose it randomly. Early JL proofs would select at random a rank k orthogonal projection

on $\mathbb{R}^Q$ [39]. This is why this dimension reduction technique is often referred to as "random projections." Conceptually, One way to think about random projections is to first apply a random rotation to $\mathbb{R}^Q$ and then read off the first k coordinates.

1.3.3 Random Projections

Random projections or mappings provide a computationally feasible method for this low-dimensional subspace embedding [45]. In a linear random projection, the high-dimensional data matrix, $\mathbf{X}$, is mapped to a lower dimensional subspace by multiplying with a random matrix $\mathbf{\Omega}$. The elements of the random matrix are samples from a probability distribution, most commonly the standard Normal distribution.

$$\mathbf{X}_c = \mathbf{\Omega} \boldsymbol{X} \tag{1.6}$$

A key concept behind random projections is that several properties of the original high-dimensional matrix $\boldsymbol{X}$ are preserved in the reduced matrix $\boldsymbol{X}_c$. This idea can be used to compute a linear function of the higher dimensional matrix $f(\mathbf{X})$ in a more efficient manner by finding an estimator $g(\mathbf{X}_c)$ such that $\mathbb{E}\left[g(\mathbf{X}_c)\right] = f(\mathbf{X})$ [46].

In structural health monitoring, the matrix $\mathbf{X}$ is composed of M time-domain measurements collected by a single sensor. Each column represents a new pitch-catch measurement, where each measurements consists of Q samples of the guided wave, taken over several milliseconds. Each measurement, $\mathbf{x}_t$, is a point that lies in a Q dimensional subspace. The samples of the measurement, x_i, are coordinates of the point in $\mathbb{R}^Q$. The coordinates are weights of orthogonal unit vectors that compose a basis for the Q dimensional subspace. When we perform the random mapping, each unit vector comprising dimension i in the original high-dimensional space is replaced with a nonorthogonal random direction $\boldsymbol{\omega}_i$ in the smaller dimensional space [45]. If the vectors $\boldsymbol{\omega}_i$ are orthogonal, the random mapping will not introduce any distortions. However, Hecht-Nielsen [47] showed that there exists a larger number of almost orthogonal directions than orthogonal directions in high-dimensional spaces. Therefore, randomly selecting directions in

a sufficiently high dimension will only provide directions that are close to orthogonal. As a result, small distortions will be introduced into similarity metrics such as correlation. Reference [45] shows that similarity metrics can be maintained within some threshold ϵ when using random matrices. The key concept behind random projections is the Johnson-Lindenstrauss lemma [39].

Random projections offer a promising solution to the growing issues of storing and processing the inevitable massive amounts of data in structural health monitoring systems. The class of linear functions are preserved in random mappings. As a result, linear signal processing methods, such as singular value decomposition, can be used on the compressed data for damage detection. In addition, random projections can significantly reduce the amount of data needed to be stored without losing much information, given that the dimension of the reduced data set is sufficiently high.

1.4 Thesis Outline

In the following chapters, we show how random projections described in the previous section can be used to dramatically reduce computational time needed for singular value decomposition-based damage detection as well as storage needs in structural health monitoring. In Chapter 2, we describe how random Gaussian matrices can be used to accomplish the two tasks previously stated for an existing data set (post processing). In Chapter 3, we extend this idea to sparse random projections that can be utilized to implement real-time damage detection in an SHM system. Finally, in Chapter 4, we discuss results found throughout this research and conclude this thesis.

CHAPTER 2

STRUCTURAL HEALTH MONITORING WITH EXISTING LARGE DATA SETS

2.1 Introduction

Structural health monitoring (SHM) seeks to give a real-time diagnosis of the state that a structure is in. However, real-time diagnostics are not always necessary. Bridges, for example, are built to endure daily loads. Determining the state of the bridge on a daily basis is not necessary. Instead, we may seek a diagnosis of the structural integrity after a set period of time or after a natural disaster, an earthquake or hurricane. In this case, a monitoring network can be placed throughout the structure to collect data. Then when a diagnosis is needed, the collected data can be recalled and analyzed offline in a postprocessing method.

As mentioned earlier, singular value decomposition [6] is an ideal damage detection technique in this scenario because these methods analyze the entire data set to extract critical trends and achieve robust damage detection. However, as the data set grows in size, this method becomes computationally expensive. Computing the full SVD of a $Q \times M$ matrix with reasonable accuracy takes $\mathrm{O}(QM \min\{Q, M\})$ floating point operations (flops). As more measurements are taken and the data matrix $\mathbf{Y}$ grows in size, the factorization of the data matrix can take a significant amount of time to compute.

In this chapter, we show that by integrating random projections [7] with singular value decomposition, the computational speed of postprocessing can be improved by a factor of 2537 times with a less than 3% reduction in accuracy. Random projection theory has been used extensively for big data processing [8]. It allows us to compress data while approximately retaining linear similarity metrics (such as correlation).

2.2 Methodologies

2.2.1 Singular Value Decomposition

In prior work, singular value decomposition SHM was used to extract structural information from guided wave data [6]. Here we will consider an $Q \times M$ data matrix where each column represents a single measurement recorded over several milliseconds and each row represents a guided wave instance over many measurements. Singular value decomposition then decomposes the matrix $\mathbf{Y}$ into three matrices such that

$$\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^H \,, \tag{2.1}$$

where $(\cdot)^H$ represents the Hermitian or conjugate transpose.

Here, and in the following sections, the $\mathbf{S}$ remains a diagonal matrix of singular values. The terms $\mathbf{U}$ and $\mathbf{V}$ are still orthogonal matrices that describe the *principal variations* in $\mathbf{Y}$. The columns of $\mathbf{V}$ describe the principal variations over days, weeks, months, or years. These variations are due to environmental and operational effects as well as damage formation and growth. The columns of $\mathbf{U}$ represent acoustic signatures that correspond to the variations in $\mathbf{V}$. Figure 2.1 graphically illustrates these matrices and their dimensions.

2.2.2 Dimensionality Reduction

As the number of measurements M grows, the storage and computational cost to perform singular value decomposition becomes prohibitive. Random projections provide an efficient storage and computational solution by reducing the dimensionality of the data. This reduction can improve the computation time of many well-established damage detection techniques. In this section, we focus on applying random projections to singular value decomposition-based methods.

We use random projections to project our high-dimensional data onto a lower dimensional subspace [7, 47] while retaining the metrics of similarity between each measurement. The key idea behind this approach is the Johnson-Lindenstrauss lemma [39]. This lemma states that a set of points in a high-dimensional subspace can be embedded into a lower dimensional subspace while nearly maintaining the distances between points.

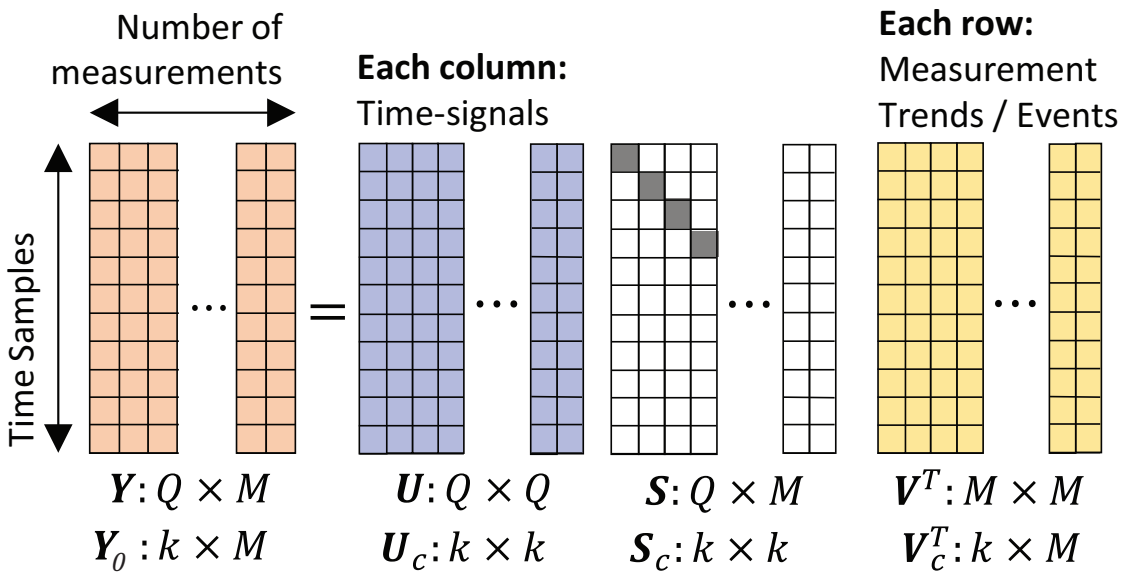

Figure 2.1: The figure illustrates the singular value decomposition of our data matrix $\mathbf{Y}$ and compressed data matrix $\mathbf{Y}_0$. Under each matrix, we show the dimensions under different conditions. The first set of dimensions illustrates the uncompressed singular value decomposition. The second set shows the compressed singular value decomposition where $k \ll Q, M$.

Consider our $Q \times M$ data matrix $\mathbf{Y}$. In general, $\mathbf{Y}$ is not full rank (i.e., the M columns of $\mathbf{Y}$ can be expressed by a linear combination of fewer than M linearly independent vectors). We can approximate our data matrix as a factorization of two lower rank matrices

$$\mathbf{Y} \approx \mathbf{B}\mathbf{Y}_0 \,, \tag{2.2}$$

where $\mathbf{B}$ and $\mathbf{Y_0}$ have dimensions of $Q \times k$ and $k \times M$, respectively. Note that the factorization is approximate since there is typically undesirable noise in the data that increases the overall rank.

Let $\mathbf{Q}$ be a matrix with orthonormal (or approximately orthonormal) columns that form an approximate length-k basis for the data matrix $\mathbf{Y}$. If we let $\mathbf{B} = \mathbf{Q}$ and $\mathbf{Y}_0 = \mathbf{Q}^H\mathbf{Y}$, then (2.2) becomes

$$\mathbf{Y}_Q \approx \mathbf{Q}\mathbf{Q}^H\mathbf{Y}, \tag{2.3}$$

The matrix $\mathbf{Y}_Q$ is a projection of $\mathbf{Y}$ onto the space spanned by the columns of $\mathbf{Q}$. If the columns of $\mathbf{Q}$ are approximately orthonormal, then $\mathbf{Q}\mathbf{Q}^H \approx I$ and $\mathbf{Y}_Q \approx \mathbf{Y}$.

Notice that $\mathbf{Y}_0$ possesses a dimension of $k \times M$. Therefore, it is a compressed version of our original data matrix. The singular value decomposition of $\mathbf{Y}_0$ can be approximated by

$$\begin{aligned} \mathbf{Y}_0 &\approx \mathbf{Q}^H \mathbf{Y} \\ &\approx \mathbf{Q}^H \mathbf{U} \mathbf{S} \mathbf{V}^H \\ &\approx \mathbf{U}_c \mathbf{S} \mathbf{V}^H \\ &= \mathbf{U}_c \mathbf{S}_c \mathbf{V}_c^H \ . \end{aligned} \tag{2.4}$$

In (2.4), the $\mathbf{V}$ matrix from the singular value decomposition of $\mathbf{Y}_0$ is approximately equal to the $\mathbf{V}$ matrix from the uncompressed data matrix $\mathbf{Y}$. The matrices $\mathbf{S}_c$ and $\mathbf{V}_c$ are "economy-size" matrices that remove the columns of $\mathbf{S}$ and $\mathbf{V}$ that correspond to the $M - k$ or more singular values equal to 0 and therefore do not contribute to $\mathbf{Y}_0$. The dimensions of these matrices are illustrated in Figure 2.1.

The results of (2.4) show that we do not need the uncompressed $Q \times M$ data matrix to perform singular value decomposition damage detection. Instead, we can process a compressed $k \times M$ matrix, where $k \ll Q$ and achieve nearly the same result. In the next two subsections, we choose $\mathbf{Q}$ to reduce both storage needs and computational costs.

2.2.3 Reducing Storage Needs with Random Projections

To reduce our storage needs, we must choose a fixed $\mathbf{Q}$ matrix with which we can multiply any new measurement. We choose $\mathbf{Q} = \mathbf{\Omega}_0$, such that

$$\mathbf{Y}_0 = \mathbf{\Omega}_0^H \mathbf{Y} \ . \tag{2.5}$$

The matrix $\mathbf{\Omega}_0$ is a $Q \times k_0$ fixed random matrix with each element defined by independent Gaussian random variables of variance of $1/\sqrt{Q}$. This choice of variance ensures that the norm of each column is approximately equal to one. This is a common approach to designing $\mathbf{\Omega}_0$, although there are many other random matrices that work. The resulting compressed data matrix $\mathbf{Y}_0$ has dimension of $k_0 \times M$.

For a $k_0 \ll Q$, the resulting matrix $\mathbf{Y}_0$ is significantly smaller than the original data matrix $\mathbf{Y}$. Any additional measurements can be multiplied by the fixed $\mathbf{\Omega}_0$

and then concatenated onto $\mathbf{Y}_0$. Based on the previous subsection, we can then apply singular value decomposition on $\mathbf{Y}_0$ to retrieve the $\mathbf{V}$ matrix. Hence, we can store and update our new compressed data matrix $\mathbf{Y}_0$ with fewer storage constraints.

Note that $\mathbf{\Omega}_0\mathbf{\Omega}_0^H\mathbf{Y}$ is not an orthogonal projection in the strict sense because the columns of $\mathbf{\Omega}_0$ are not strictly orthogonal [7]. However, when reducing from sufficiently high-dimensional spaces, the columns are orthogonal in expectation. In practice, this means that we can achieve better compression with greater accuracy if we can construct a $\mathbf{Q}$ matrix with strictly orthogonal columns. In the following subsection, we create this $\mathbf{Q}$ matrix. Note though that the next $\mathbf{Q}$ matrix cannot be directly used for storage improvement because the matrix will change as we add measurements into our data.

2.2.4 Reducing Computation with Random Projections

As previously described, singular value decomposition-based damage detection traditionally computes the full decomposition of our $Q \times M$ data matrix $\mathbf{Y}$. However, as Q (the number of time samples) and M (the number of measurements) grow, the full decomposition becomes computationally intractable. Singular value decomposition has a computational complexity of $\mathcal{O}(\min(QM^2, Q^2M))$. In the previous subsection, we premultiplied the data matrix with the random matrix $\mathbf{\Omega}_0$ to reduce Q to size k_0. Therefore, if this step reduces storage needs by 100 times, it also improves our computational speed by 100 to $10,000$ times.

In this section, we further reduce the computational cost by choosing a more effective $\mathbf{Q}$ matrix for (2.4). To create this $\mathbf{Q}$ matrix, we further process the compressed data matrix

$$\mathbf{Y}_1 = \mathbf{Y}_0\mathbf{\Omega_1}\,, \tag{2.6}$$

where $\mathbf{\Omega_1}$ is an $M \times k_1$ random matrix. The matrix $\mathbf{Y}_1$ has dimensions of $k_0 \times k_1$. As with $\mathbf{\Omega}_0$, the elements in $\mathbf{\Omega}_1$ are independent, Gaussian random variables and the columns are orthonormal in expectation. We then find a new $\mathbf{Q}$ matrix (now with orthonormal columns) by performing a QR decomposition on $\mathbf{Y}_1$, such that

$$\mathbf{Y}_1 = \mathbf{QR} \, . \tag{2.7}$$

The matrix $\mathbf{Q}$ has a length of $k_0 \times \min(k_1, k_0)$. Each column in $\mathbf{Q}$ represents part of an orthonormal basis for $\mathbf{Y}_1$. Note that the computational complexity for the QR decomposition is $\mathbf{O}(k_1^3)$, which is not significant when k_1 is sufficiently small.

We create our final compressed data matrix $\mathbf{Y}_c$ with dimensions of $k_1 \times M$ (assuming $k_1 < k_0$) such that

$$\mathbf{Y}_c = \mathbf{Q}^H \mathbf{Y}_0 \, . \tag{2.8}$$

We then apply singular value decomposition on $\mathbf{Y}_c$ to retrieve the approximate $\mathbf{V}$ matrix. Since $\mathbf{Q}$ has strictly orthonormal columns, this process should reduce the k_0 column length of $\mathbf{Y}_0$ to k_1 and further reduce the computational burden of singular value decomposition.

2.3 Experimental Setup

We demonstrate the effectiveness of random projections with a collection of guided data from an aluminum plate. The purpose of this experiment is to compress and process the data very efficiently. Therefore, for simplicity, this preliminary experiment does not include significant temperature variations.

We consider a 50.8 cm by 50.8 cm aluminum plate with circular 10 mm diameter PZT (lead zirconate titanate) transducers at opposite corners of the plate. From one transducer, we transmit a 0.5 ms linear chirp with a center frequency of 225 kHz and a bandwidth of 350 kHz. The second transducer receives and stores this signal after travelling through the plate. Before processing, the data was filtered with a Gaussian filter at a center frequency of 20 kHz and a bandwidth of 20 kHz. The low frequencies (i.e., under approximately 100 kHz) were generally more sensitive to the masses on the plate.

The transmit-receive process was repeated every 10 minutes over one week. In total, we collected 1447 measurements. Just before measurement 62 (hour 10.3), an uncoupled 8.9 cm by 3.81 cm aluminum piece was placed on the plate. Just before measurement 474 (hour 79), another heavier, but still uncoupled, 10.2 cm by 10.2 cm aluminum piece was placed on the plate. Just before measurement 1065

(hour 177.5), a pair of grease-coupled circular magnets with 1.9 cm diameters and separated by approximately 5 cm were placed on the top surface of the plate. A 7.93 cm by 5.6 cm uncoupled steel plate was placed on the back surface of the plate to secure each of the magnets. Figure 2.2 illustrates the approximate locations.

2.4 Results

Performing singular value decomposition on the original, uncompressed data requires approximately 8.5 s, or 8500 ms. Figure 2.3(a) illustrates the computational time to perform a singular value decomposition on our compressed data $\mathbf{Y}_c$. The horizontal axis illustrates the initial dimension reduction to a length of k_0 (for improved storage) and each line corresponds to a second reduction to length k_1 (for computational improvement). The figure shows an approximate linear relationship between k_0 and computation time until $k_0 = k_1$. After this point, computation time is entirely determined by the choice of k_1. Notice that by reducing our dimension to $k_1 = 10$ with $k_0 > k_1$, the computation time reduces to approximately

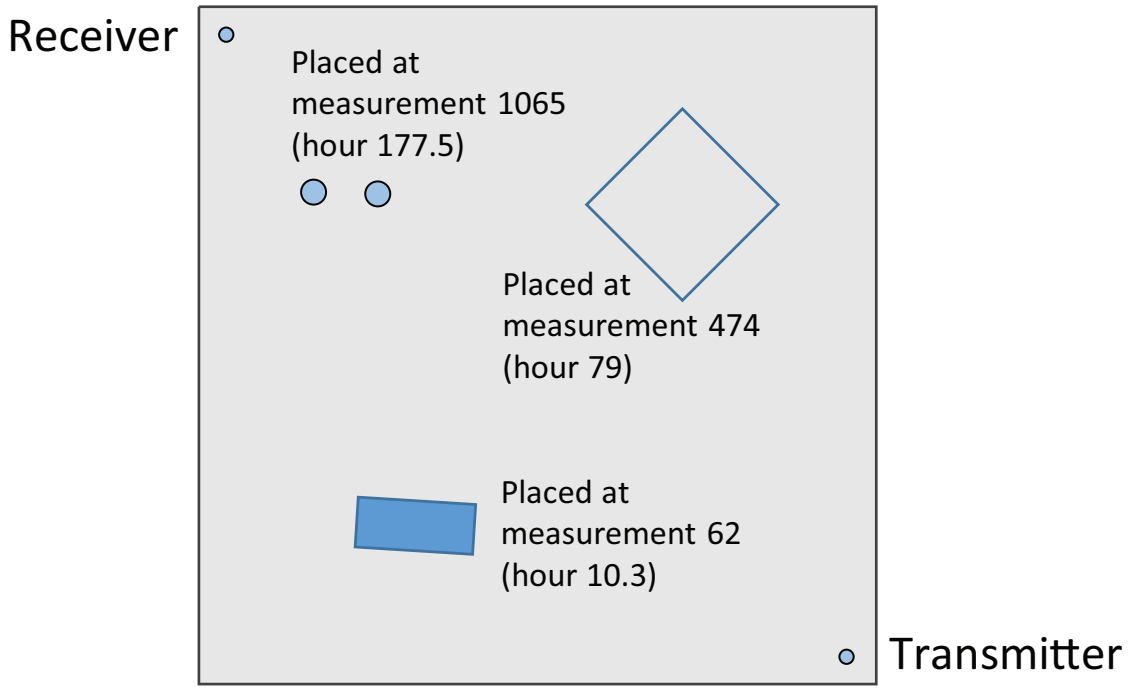

Figure 2.2: The figure illustrates the experimental setup and the approximate positions of the masses that we places over time. The grease-coupled magnets placed during measurement 1065 create the most significant variations.

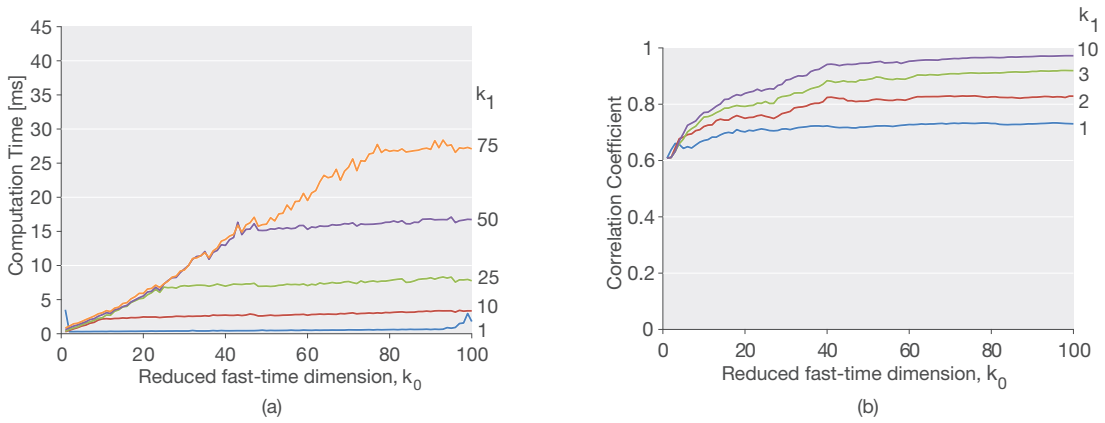

Figure 2.3: Effects of low rank reduction on computation time and correlation. (a) The measured computation times for computing the singular value decomposition of our compressed data set $\mathbf{Y}_c$ for different combinations of k_0 and k_1. (b) The correlation coefficient between singular vectors from the compressed data set $\mathbf{Y}$ and the uncompressed data set $\mathbf{Y}$ for different combination of k_0 and k_1. A correlation coefficient of 1 indicates no change in the singular vectors.

3.35 ms, a 2537 times improvement in speed.

Figure 2.3(b) illustrates the correlation coefficient between the rows of $\mathbf{SV}^T$ from the uncompressed data $\mathbf{Y}$ and the same vectors generated from the compressed data $\mathbf{Y}_c$. A correlation coefficient of 1 indicates a perfect recovery. We see a gradual improvement in accuracy as both k_0 and k_1 increase. As k_1 increases beyond $k_1 = 10$, we see only minor improvements. This plot illustrates the benefits of further reducing our data's dimension from k_0 to k_1. Reducing the data initially to $k_0 = 10$ would only achieve a correlation coefficient of 0.76. Yet, choosing $k_0 = 100$ and then $k_1 = 10$ achieves a correlation coefficient of 0.97 with nearly the same computational expense, as shown in Figure 2.3(a).

Figure 2.4 illustrates the first two singular vectors taken from $\mathbf{V}$. The first singular vector represents overall changes in the signal over the course of the experiment. The second singular vector isolates the most substantial event in our data set – the addition of the grease-coupled magnets at measurement 1065 (after approximately 177.5 hours of operation). This confirms that we can use singular value decomposition to isolate simulated damage events in the data set.

Overall, our results show that if we choose to reduce the data with $k_0 = 100$ and $k_1 = 10$, we would achieve a 100 times improvement in storage efficiency

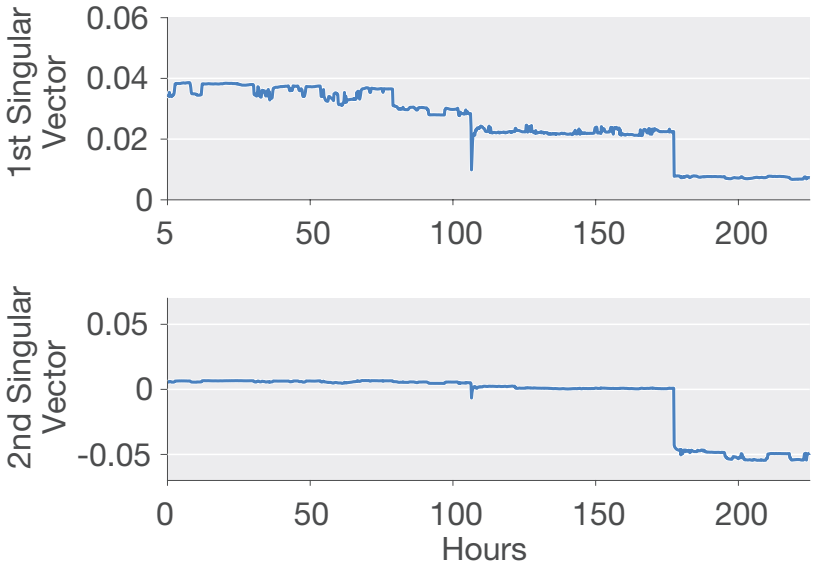

Figure 2.4: The first two singular vectors of **V** from taken from the uncompressed data set. The second singular vector isolates the most significant event from the experiment.

and a 2537 times reduction in computation time with only a 3% reduction in the correlation coefficient.

2.5 Discussion

This chapter demonstrates how to use random projections to improve storage and computational efficiency of guided wave structural health monitoring damage detection algorithms. We achieved greater than 2537 times computational improvement after applying random projections to a 1447 measurement data set. Note that this approach scales well with the number of measurements. That is, if we collected 10 times the number of measurements, we would expect a 10 times improvement in computation time. As structural health monitoring systems grow, these algorithms will be essential to data analysis.

CHAPTER 3

STRUCTURAL HEALTH MONITORING WITH LARGE DATA SETS FOR REAL-TIME SYSTEMS

3.1 Introduction

Structural health monitoring (SHM) seeks to give a real-time diagnosis of the state of the structure. However, as the SHM system ages, the amount of data collected (and subsequently analyzed) grows in size. As a result, many successful damage detection methods are bottlenecked by the number of dimensions associated with the growing number of measurements. Many damage detection techniques, namely singular value decomposition (SVD), scale poorly as the size of the dimensions of the data set grow. For a $M \times Q$ matrix, the SVD factorization has a computational complexity of $O(\min(QM^2; Q^2M))$. Therefore, as the quantity $\min(QM^2; Q^2M)$ grows larger, SVD becomes a less viable damage detection option for a real-time system. In this chapter, we use Johnson-Lindenstrauss embeddings to overcome this challenge and maintain SVD's standing as a viable damage detection method, even in a real-time system.

There are three main challenges to consider when designing a real-time system, referred to here as the 3S challenge.

- *Size* - How much data is collected.
- *Speed* - How fast can we process data.
- *Scalability* - How well does the process scale.

In the previous chapter, we addressed the first challenge, *size*. Namely, we showed that we can use Gaussian matrices to perform JL-embeddings and reduce the size

of the data. In this chapter, we will address the second challenge, *speed*. Gaussian matrices are dense matrices. When a Gaussian matrix $\mathbf{\Omega} \in \mathbb{R}^{Q\times k}$ is used to perform JL-embeddings and the dimension Q is large, a lot of memory and computation is needed to perform the embedding. This slows down analysis methods [48]. Due to this fact, we adopt a method of using a *sparse* random matrix. Sparse matrices will allow us to perform the embeddings with only a fraction of the data, speeding up embedding computations. Reference [49] showed that sparse matrices can be used to perform JL-embeddings with little loss in accuracy. Furthermore, this is a simple matrix to create as it requires only a uniform random number generator. In this chapter, we show that we can use a sufficiently sparse matrix to improve embedding times, thereby reducing computation times for damage detection methods when dealing with very large dimensions. Due to these fast embeddings, we can use SVD as a viable damage detection technique for real-time SHM systems, as opposed to postprocessing.

3.2 Methodology

3.2.1 Low-dimensional Subspace Embeddings

Here we will consider a transpose of the $M \times Q$ matrix $\mathbf{Y}$ dealt with in Chapter 2; however, in this setting, it will be useful to consider this as a set of M points in a Q dimensional space. Recall that the Johnson-Lindenstrauss lemma states that a set of points in a high-dimensional space can be embedded into a lower dimensional space where the pair-wise (euclidean) distances, between points, can be approximately preserved with high probability. One important property of the JL lemma, pointed out in Chapter 1, is the fact that the dimension into which we wish to embed our points only depends on the number of pairwise distances we wish to preserve (the distance between any two columns in $\mathbf{Y}$). The more data we collect, the higher k must be.

This poses an unfortunate limitation on the method of random projections. In the previous chapter, we showed that given a data set, we can reduce both dimensions Q and M, thereby, significantly improving computation times and storage needs. However, in a real-time monitoring system, this is not possible

as we do not have all the measurements. Instead we collect measurements one at a time and therefore, the dimension M grows in size over time. As a result of this, the more data that is collected, the slower SVD will be at performing damage detection. However, random projections are still useful in this scenario. Random projections can be used to decrease the dimension Q, which can be quite large. As was discussed in Chapter 1, this thesis addresses guided wave structural health monitoring. In guided wave SHM, sensors will transmit a Lamb wave and the remaining sensors will capture the received transmission. The physical nature of the structure and Lamb waves will effect the dimension of the measurements. Lamb waves contain high-frequency components that must be sampled at sufficiently high rates in order to avoid aliasing. Therefore, the higher the frequency content of the Lamb wave, the higher the dimension Q of the measurements will be. Furthermore, the dimension Q is dependent upon the spacing between sensors. The farther sensors are, the longer the sensors must sample as the Lamb wave traverses the structure, thereby increasing the dimension Q as well. From this, we can see that the dimension Q can be quite large and reducing this dimension will provide significant improvements in computing the SVD of the data set.

3.2.2 Gaussian Subspaces

The original work of finding JL embeddings approached the problem from a geometric viewpoint. While although a geometric interpretation may be insightful, it can lead to a rather cumbersome analysis. Dasgupta and Gupta substantially simplified the proof for finding JL embeddings by approaching the problem for a probabilistic viewpoint [50]. First, let us define a k-dimensional *random* subspace $\mathbf{\Omega} \in \mathbb{R}^{Q \times k}$ by choosing k random vectors independently from the standard Normal distribution, where each element in a random vector is chosen according to $\omega_i \sim \mathcal{N}(0,1)$. Then, let z_i denote the ith coordinate of a projection in the random subspace by taking the inner product of a fixed unit vector, $\mathbf{y} \in \mathbb{R}^{Q}$, with the ith random vector. According to the 2-stability of the Gaussian distribution: for any real numbers $y_1, y_2, ..., y_Q$, if $\boldsymbol{\omega}$ is a vector of independent Normal random variables then [51]

$$\begin{aligned} z &= \sum_{i=1}^{d} y_i \omega_i \\ &= \boldsymbol{\omega}_i^T \mathbf{y} \\ &\sim \mathcal{N}(0, ||\mathbf{y}||^2) \end{aligned} \tag{3.1}$$

From (3.1) we see that each coordinate in the projected vector is also a Gaussian random variable. Furthermore, we can interpret the mapping $\mathbf{z} = \boldsymbol{y\Omega}$ as mapping $\mathbf{y}$ from $\mathbb{R}^Q$ to $\mathbb{R}^k$, where each new coordinate of $\mathbf{y}$ in $\mathbb{R}^k$ is obtained by taking the inner product of $\mathbf{y}$ with a column in $\boldsymbol{\Omega}$. Then, the squared length of the vector $\mathbf{z}$ follows the Gamma distribution which possesses strong concentration bounds [52]. Therefore, we can bound the length of the vector $\mathbf{y}$ in the random mapping.

For a complete proof of (1.5) where the projection matrix has elements drawn from a Gaussian distribution, the reader is directed to [53–55]. Below, we will try to provide some high-level intuition on Gaussian random projections. Consider the mapping $\boldsymbol{\Omega} : \mathbb{R}^Q \to \mathbb{R}^k$, where $k \ll Q$. Assume each element of $\boldsymbol{\Omega}$ is defined by an independent Gaussian random variable with zero mean and variance $\sigma^2 = 1$ and the signal $\mathbf{y} \in \mathbb{R}^Q$. We have that

$$\begin{aligned} E\left[||\boldsymbol{y\Omega}||_2^2\right] &= E\left[(\boldsymbol{y\Omega})(\boldsymbol{y\Omega})^*\right] \\ &= E\left[\boldsymbol{y\Omega\Omega^* y^*}\right] \\ &= \mathbf{y} E\left[\boldsymbol{\Omega\Omega^*}\right] \mathbf{y}^* \\ &= \mathbf{y}\mathbf{I}\mathbf{y}^* \\ &= ||\mathbf{y}||_2^2 \end{aligned}$$

We can extend this short proof to preserving distances by letting the vector $\mathbf{y} = \mathbf{u} - \mathbf{v}$, where $\boldsymbol{u}, \boldsymbol{v}$ are two points in $\mathbb{R}^Q$ and repeating the same proof above. Furthermore, random projections preserve inner products, which can be shown using a similar proof to the one above.

3.2.3 Non-Gaussian Matrices

In an effort to make random projections easier to use in practice, Achlioptas [49] presented a much simpler method for creating the random projection and

performing the embeddings. The main results in [49] show that spherical symmetry (the projection onto $\mathbb{R}^k$ maintains the same distribution as the matrix $\mathbf{\Omega}$) is not necessary. In fact, it is only necessary for the square of the projections to be concentrated about the norm of the fixed vector, in expectation. This can be accomplished by drawing the elements of the random matrix independently from a zero mean distribution.

As we saw earlier, the inner product between the fixed vector $\mathbf{y} \in \mathbb{R}^Q$ with rows of $\mathbf{\Omega} \in \mathbb{R}^{Q \times k}$ gives the coordinates of the projection in $\mathbb{R}^k$. Furthermore, the squared sum of the coordinates gives the length of the projection. We can think of the inner product of $\mathbf{y}$ with each row of $\mathbf{\Omega}$ as effectively giving an estimator for the length of the projection that we then average over k estimates (one for each row). In the previous chapter, we performed the projection using Gaussian random variables. However, it turns out that so long as the values of $\mathbf{\Omega}$ are sampled from a zero mean distribution, we will obtain an unbiased estimate on the length of the vector $\mathbf{y}$ [49, 56]. Intuitively, we can describe this result as

$$\mathbf{z}_i = \frac{1}{k}\boldsymbol{y_i}\mathbf{\Omega}$$

$$\begin{bmatrix} \mathbf{y}_i\boldsymbol{\omega}_1 \\ \mathbf{y}_i\boldsymbol{\omega}_2 \\ \vdots \\ \mathbf{y}_i\boldsymbol{\omega}_k \end{bmatrix}^T = \begin{bmatrix} y_1 & y_2 & \dots & y_Q \end{bmatrix} \begin{bmatrix} | & | & & | \\ \boldsymbol{\omega}_1 & \boldsymbol{\omega}_2 & \dots & \boldsymbol{\omega}_k \\ | & | & & | \end{bmatrix}$$

$$\begin{aligned} E[||\mathbf{z}_i||_2^2] &= E[\frac{1}{k}||\mathbf{y}_i\mathbf{\Omega}||_2^2] \\ &= \frac{1}{k}\mathbf{y}_i E[\mathbf{\Omega}\mathbf{\Omega}^T]\mathbf{y}_i^T \\ &= \frac{1}{k}\mathbf{y}_i\mathbf{\Sigma}\mathbf{y}_i^T \\ &\approx \frac{1}{k}\sum_{j=1}^{k}||\mathbf{y}_i||_2^2 \text{var}(\boldsymbol{\omega}_j) \\ &\approx ||\mathbf{y}_i||_2^2 \end{aligned} \tag{3.2}$$

From (3.2), we notice a few properties. First, so long as the random vector $\boldsymbol{\omega}$ is zero mean, we will obtain an unbiased estimator for the length of $\mathbf{y}$. Second,

the accuracy of the estimate depends on the variance of the random vector ω. Therefore, the distribution selected to create the mapping $\mathbf{\Omega}$ plays an important role in the accuracy of the estimator. Lastly, the closer the random matrix is to being orthogonal, the closer the off diagonal elements in the covariance matrix $\mathbf{\Sigma}$ will be to zero and the better each estimate will be. Furthermore, so long as each column of $\mathbf{\Omega}$ is independent and zero mean, we can apply the central limit theorem and take enough projections to get a satisfactory estimate of length. Where the number of projections needed depends on the variance of the estimate [49]. Furthermore, we can bound the concentration by applying the Law of Large Numbers.

$$\text{Prob}\left(\|\frac{z_1 + z_2 + ... + z_k}{k} - E[\mathbf{z}]\| \geq \epsilon\right) \leq \frac{\text{var}(\mathbf{z})}{k\epsilon^2} \tag{3.3}$$

From 3.3, it is apparent that the concentration, and hence the distortion, of the lengths of the projected vectors are dependent on the variance of the distribution used to create the random mapping $\mathbf{\Omega}$. Therefore, we are free to choose distributions other than the Gaussian distribution.

The computation time to perform the embeddings can be greatly improved by selecting elements from the set $\mathbf{\Omega}_{ij} = \{-1, +1\}$ with probability $\left\{\frac{1}{2}, \frac{1}{2}\right\}$. This is a zero mean distribution with unit variance and will therefore act as an unbiased estimator for the length of the original vector. Also, we know from [47] that there exists more *almost* orthogonal vectors than orthogonal vectors in a high-dimensional space. The covariance matrix of $\mathbf{\Omega}$ will then be close to the identity matrix, and hence the length of the projected vector will be an accurate approximation of the original vector's length. By drawing from this distribution, there are no floating point operations needed in the embedding, saving significant computations and memory in the process. The projection can be completed much faster by drawing the elements of $\mathbf{\Omega}$ independently from a *sparse* distribution.

3.2.3.1 Sparsity

As was previously shown, it is not necessary to use the standard Normal distribution to create the random matrix for low-dimensional embeddings. Instead, it is possible to project onto a matrix where elements of $\mathbf{\Omega}$ are drawn from $\mathbf{\Omega}_{ij} \in$

$\{-1, +1\}$ [57]. Both of these choices are simple to construct but are dense matrices. This is generally undesirable. For example, embedding into a lower subspace using Gaussian matrices requires many floating point operations and a large amount of memory to perform the computations. Therefore, it would be beneficial to find a distribution other than Gaussian.

Using a projection matrix where the elements are defined by the Bernoulli distribution, $\mathbf{\Omega}_{ij} \in \{-1, +1\}$, is simpler to create and the embeddings using this distribution do not require any floating point operations. However, we can further reduce our computation because the projection matrix need not be dense. Let $\mathbf{\Omega} : \mathbb{R}^Q \to \mathbb{R}^k$ where the elements are defined by independent random variables drawn from the *sparse* Bernoulli distribution defined as [56]

$$\mathbf{\Omega}_{ij} = \sqrt{s} \begin{cases} +1 \text{ with probability} & \frac{1}{2s} \\ 0 \text{ with probability} & 1 - \frac{1}{s} \\ -1 \text{ with probability} & \frac{1}{2s} \end{cases}$$

By creating the matrix $\mathbf{\Omega}$ to be a sparse matrix, the computation time for the JL embeddings can be sped up by a factor of s because there are only $\frac{1}{s}$ non zero entries per row[49, 56]. Since the multiplicative factor $\sqrt{s}$ can be delayed, the JL embedding does not require any floating point operations, providing another improvement to computational speed. Even with the use of a sparse projection matrix, we lose surprisingly little in terms of preserving pair-wise distances in the embedded subspace.

3.2.3.2 Real-time Processing

Due to the significant speed up in the time needed to perform the embeddings, sparse random matrices are a prime candidate to assist in real-time damage detection. In order to reduce computations, we choose a *sparse* Bernoulli distribution (defined above) to create the projection matrix $\mathbf{\Omega}$.

When we initialize our structural health monitoring system, we begin with an empty "compressed" data matrix, $\mathbf{Y}_0 \in \mathbb{R}^k$. Periodically, sensors will interrogate the structure providing a new measurement $\mathbf{y}_t \in \mathbb{R}^Q$ at time t. This new measurement is then mapped into $\mathbb{R}^k$ using the sparse Bernoulli matrix defined above and

then added as the t^{th} row vector in the "compressed" data matrix $\mathbf{Y}_0$.

$$\begin{aligned} \mathbf{y}_{0,t} &= \mathbf{y}_t\mathbf{\Omega} \\ \mathbf{Y}_{0,t} &= [\mathbf{Y}_{0,t-1}^T \quad \mathbf{y}_{0,t}^T]^T \end{aligned} \tag{3.4}$$

Equation (3.4) shows how we can compress measurements in real time as we receive them. It also shows a data matrix that is growing row-wise with each new measurement. This "compressed" data matrix resides in $\mathbb{R}^k$ as opposed to the original measurements, which reside in $\mathbb{R}^Q$, where $k \ll Q$. Because of this, when the number of measurements is comparable to the size of k, this will provide significant computational improvements when computing the singular value decomposition; this is shown below in the Results section.

Recall the treatment of dimensionality reduction given in Chapter 2. We showed that we can successfully compute the singular value decomposition of the compressed data matrix. However, in Chapter 2, we found a rank k orthonormal basis $\mathbf{Q}$ for the data matrix by randomly mapping it to $\mathbb{R}^k$, and orthogonalizing the mapped vectors. We then restricted the data to this basis $\mathbf{B} = \mathbf{Q^H Y}$ and computed SVD for this result. This provided an approximation for the SVD of the original data set.

$$\begin{aligned} \mathbf{Y}_0 &= \mathbf{Y\Omega} \\ &= \mathbf{USV}^T\mathbf{\Omega} \\ &= \mathbf{US}\tilde{\mathbf{V}}^T \end{aligned} \tag{3.5}$$

Here, we rely on the results of [47], proving there exists more *almost* orthogonal vectors in high dimensions than orthogonal vectors. Sampling random vectors from the *sparse* Bernoulli distribution should give orthogonal vectors in expectation, that $E[\mathbf{\Omega\Omega}^T] \approx \mathbf{I}$.

$$\begin{aligned} E[\boldsymbol{Y_0\Omega^T}] &= E[\mathbf{US}\tilde{\mathbf{V}}^T\mathbf{\Omega}^T] \\ &= \mathbf{USV}^T E[\mathbf{\Omega\Omega}^T] \\ &\approx \mathbf{USV}^T \end{aligned} \tag{3.6}$$

Due to the fact that we are storing new measurements as rows in (3.4), the roles of $\mathbf{U}$ and $\mathbf{V}$ switch. Therefore, we are interested in the columns of $\mathbf{U}$ in order to

observe slow-time trends, such as damage, in the data. The last step (3.6) is to obtain an approximation for $\mathbf{V}$, which is unnecessary since we are interested in the columns of $\mathbf{U}$. Therefore, we see that we can apply singular value decomposition on the compressed data, $\mathbf{Y}_0$, to retrieve the $\mathbf{U}$ for real-time damage detection.

3.3 Results

We use the same experimental setup established in Chapter 2. The transmit-receive process was repeated every 10 minutes over one week. In total, we collected 1447 measurements. Just before measurement 62 (hour 10.3), an uncoupled 8:9 cm by 3.81 cm aluminum piece was placed on the plate. Just before measurement 474 (hour 79), another heavier, but still uncoupled, 10.2 cm by 10.2 cm aluminum piece was placed on the plate. Just before measurement 731 (hour 121.83), the heavier mass was removed. Just before measurement 1065 (hour 177.5), a pair of grease-coupled circular magnets with 1.9 cm diameters and separated by approximately 5 cm were placed on the top surface of the plate. A 7.93 cm by 5.6 cm uncoupled steel plate was placed on the back surface of the plate to secure each of the magnets.

Performing singular value decomposition on the uncompressed data matrix took approximately 8 seconds. Figure 3.1(a) shows the computation time required to perform singular value decomposition using a sparse projection matrix. The horizontal axis corresponds to the sparsity of the projection matrix. This can be thought of as a percentage of the matrix consisting of zero elements. Each line in 3.1(a) corresponds to a dimension of a lower subspace into which the data is embedded. For $s = 100$ and $k = 50$, the computational time required to process the entire data set is roughly 19 ms, a 450 times improvement over computing SVD on the uncompressed data set. Figure 3.1(b) shows the correlation between the rows of $\mathbf{SV}^T$ of the original data set and the compressed data set. Again, each line in 3.1(b) corresponds to the dimension of the lower dimensional subspace. From the figure, there is almost perfect correlation between the original data and the compressed data for $k = 50$. Note that, the correlation seems to be independent of the sparsity of the matrix. According to [56], We can achieve a sparsity of upwards of $s =$

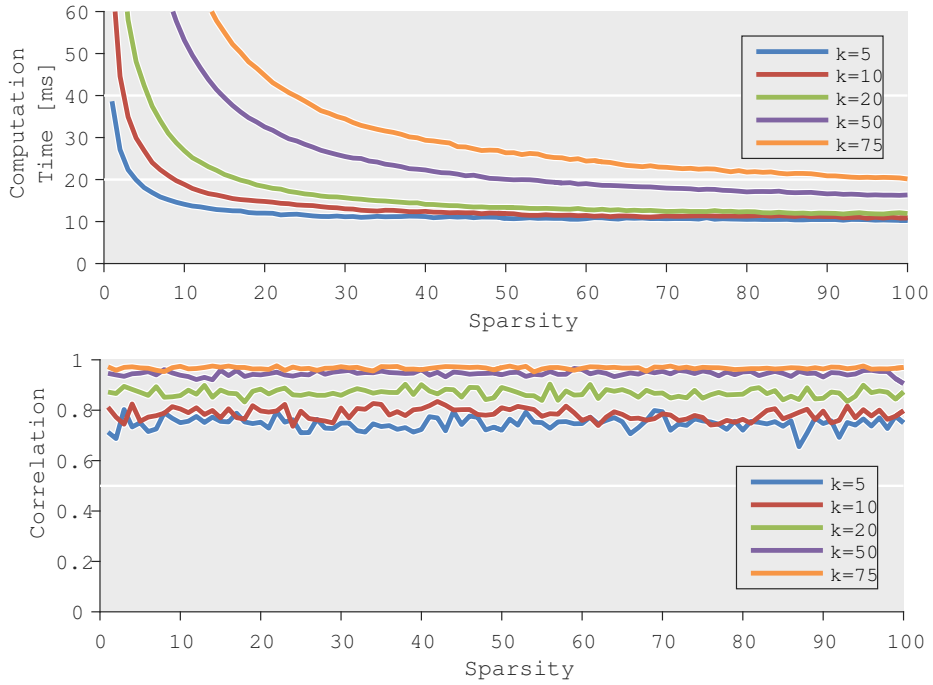

Figure 3.1: Effects of sparsity on computation time and correlation. (a) The measured computation times for computing the singular value decomposition of our compressed data set for different combinations of sparsity and reduced dimension, k. (b) The correlation coefficient between singular vectors from the compressed data set and the uncompressed data set for different combination of sparsity and reduced dimension k. A correlation coefficient of 1 indicates no change in the singular vectors.

$\frac{Q}{\log(Q)}$ (however, they recommend a less aggressive sparsity such as $s = \sqrt{Q}$) before we begin to see a decrease in accuracy. The computation time in Figure 3.1(a) appears to be inversely exponential and therefore, we see diminishing returns on the computational savings as the sparsity gets large.

Figure 3.2 illustrates the computational improvement of using a *sparse* Bernoulli distribution (with sparsity $s = \frac{Q}{\log(Q)} \approx 1000$) over a Gaussian distribution to create $\mathbf{\Omega}$. The horizontal axis of Figure 3.2 is the correlation between the rows of $\boldsymbol{SV}^T$ of the original data set and the compressed data set. The vertical axis shows the computational improvement, which is the ratio of the time it takes to compute the SVD of the original uncompressed data set to the compressed data set. Figure 3.2 shows that for a correlation of 0.97, the *sparse* Bernoulli achieves roughly a 115% increase in computational improvement, thus giving a clear indication

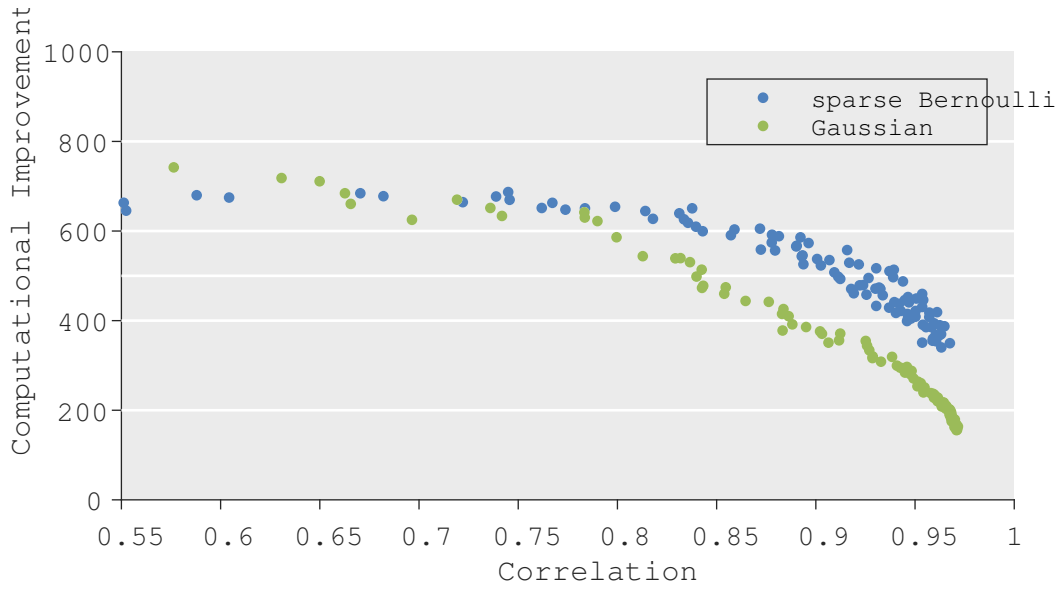

Figure 3.2: Comparison of embedding techniques using *sparse* Bernoulli distribution and Gaussian distribution to create $\mathbf{\Omega}$.

that a sparse matrix can achieve high of accuracies for damage detection, while significantly outperforming traditional random matrices.

Figure 3.3 shows the first singular vectors of the compressed data and the original data for $s = 1000$. The first singular vector for the compressed data is shown on top and the first singular vector for the original data is the bottom plot. The singular vector for the compressed data set, even with a large sparsity, looks identical to the the original singular vector. From this, we can tell exactly when damage was introduced to the system by observing the "steps" in the singular vector.

3.4 Discussion

Singular value decomposition scales poorly as the size of the data set grows. This chapter demonstrates that singular value decomposition can be a viable option for real-time damage detection when embedded into a lower dimensional subspace using *sparse* random projections. This sparse random projection consists of {+1,0,-1} and is easy to create using a uniform random generator. The benefit of a sparse matrix over a Gaussian matrix is no costly floating point operations

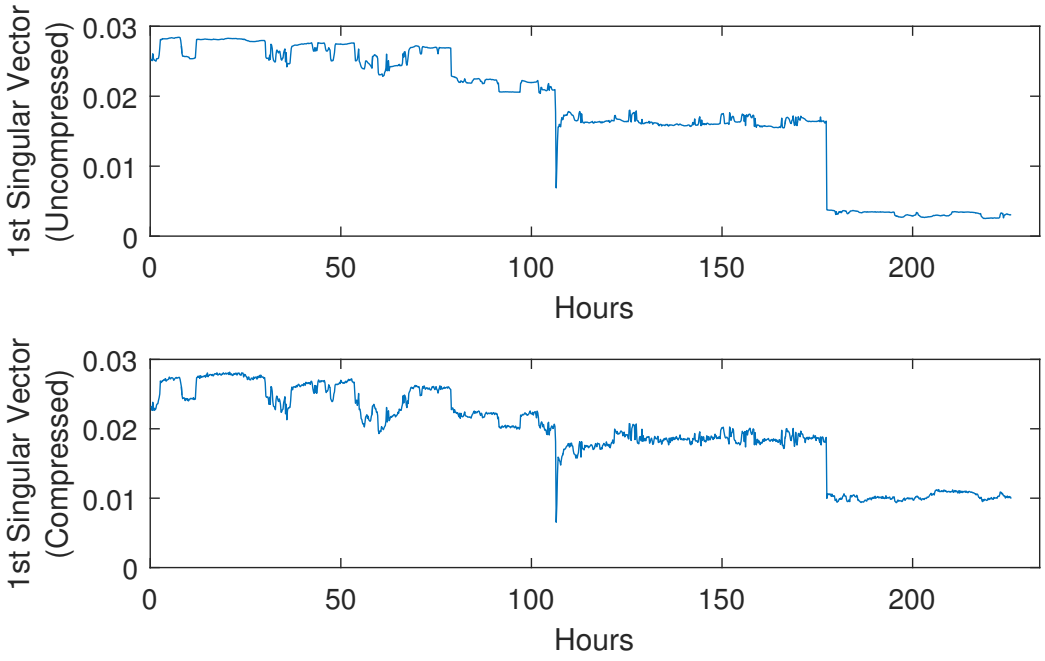

Figure 3.3: The 1st singular vectors for the compressed data set and the original data set for $s = 1000$.

are needed in the embedding, and approximately only $\frac{1}{s}$ data points are needed, saving memory during the embedding. We lose surprisingly little, in terms of accuracy, when using sparse matrices. We showed there remains an almost perfect correlation between the compressed data and the original data, while introducing a 450 fold improvement over the uncompressed data in computation time.

CHAPTER 4

CONCLUDING REMARKS AND FUTURE WORK

4.1 Summary

In this thesis, we proposed random matrices as a tool for dealing with the ever-increasing size of data sets collected during structural health monitoring (SHM). Singular value decomposition is a robust tool for damage detection; however, it scales poorly with the size of the data. Projecting the large data set onto a random smaller subspace allows for computational improvements when computing the singular value decomposition of the data. Furthermore, the Johnson-Lindenstrauss lemma demonstrates that so long as the reduced subspace is sufficiently large, distance between points in the data set can be approximately maintained. Therefore, these significant computational improvements can be achieved with little loss in accuracy.

We began with a brief literature review in the field of SHM, particularly guided wave SHM and a few damage detection frameworks, namely baseline subtraction and singular value decomposition (SVD). SVD's ability to extract critical trends across a data set makes it an excellent damage detection method that is robust to temperature variations. We then provided a brief description of popular data compression tools such as principal component analysis and random projections. We show that random matrices can be applied to SHM data sets to boost computational speed on a large data set or when a data set needs to be reduced in order to save storage cost. Thus, our approach keeps SVD as a viable damage detection technique in a real-time SHM system.

Chapter 2 demonstrated that, when we have existing data sets, we can use random projections to improve storage and computational efficiency of guided wave

structural health monitoring damage detection algorithms. We achieved a greater than 2537 times computational improvement over SVD on the uncompressed data after applying random projections to a 1447 measurement data set. Furthermore, we noted that this approach can scale well with the number of measurements. That is, if we collected 10 times the number of measurements, we would expect a 10 times improvement in computation time. As structural health monitoring systems grow, these algorithms will be essential to postprocessing data analysis.

In Chapter 3, we further iterated the point that singular value decomposition scales poorly as the size of the data set grows. We then demonstrated that singular value decomposition can be viable option for real-time damage detection when embedded into a lower-dimensional subspace using random projections. We also pointed out that while creating the random matrix from a Gaussian distribution provides a simple and intuitive analysis, it also results in a dense matrix that slows down embedding times. So long as the elements of the random matrix are drawn independently from a zero mean distribution, we can successfully use random projections for SHM applications. This allows us to pursue other distributions for the embedding that will be computationally feasible for performing the embeddings in real time.

Furthermore, we used a sparse random projection consisting of {+1, 0, -1} with probability $\{\frac{1}{2s}, 1 - \frac{1}{s}, \frac{1}{2s}\}$ to perform embeddings. This distribution is easy to create using a uniform random number generator. Using this distribution to create the random matrix drastically improved the computational time needed to perform the embeddings. The benefit of a sparse matrix over a Gaussian matrix is that no costly floating point operations are needed in the embedding, and approximately only $\frac{1}{s}$ data points are needed, saving memory during the embedding. In addition, we lose surprisingly little in terms of accuracy. We showed there remains an almost perfect correlation between the compressed data and the original data, while introducing a 450 fold improvement in computation time compared to computing the SVD on the original data.

4.2 Future Work

We provide here a brief list of suggested work that would elaborate and extend the work described in this thesis.

As was mentioned earlier, there are three main challenges presented to a real-time system: size, speed, and scalability. The work done in this thesis has addressed the first two challenges, namely size and speed. However, this work should be extended to a network of sensors to ensure random projection dimension reduction technique can be scaled. In practice, a network of sensors is needed to implement any sort of sophisticated structural health monitoring system. We believe random projections have the capability to scale well with the complexity of SHM systems.

Structural health monitoring systems need to be able to detect when a structure is damaged. Furthermore, SHM systems need to be able to distinguish between multiple sources of damage in order to successfully operate. There are a limited number of algorithms that can successfully perform this task of damage separation. These algorithms need to be tested on data sets once they have been projected onto a random subspace.

Finally, guided wave-based SHM is susceptible to environmental conditions that can drastically alter the received signals. Environmental compensation algorithms need to be created to work on data sets after they have been projected onto a random subspace. Johnson-Lindenstrauss lemma states that random projections can approximately maintain *Euclidean* distances. Therefore, any environmental compensation algorithm will most likely need to be able to perform on euclidean distances.

REFERENCES

[1] "Pipeline incidents 20 year trend," U.S. Department of Transportation and Hazardous Materials Safety Administration, Tech. Rep., 2015. [Online]. Available: https://hip.phmsa.dot.gov/analyticsSOAP/saw.dll?Portalpages

[2] D. Balageas, C.-P. Fritzen, and A. Güemes, *Structural health monitoring*. Wiley Online Library, 2006.

[3] A. Raghavan and C. E. Cesnik, "Review of guided-wave structural health monitoring," *Shock and Vibration Digest*, vol. 39, no. 2, pp. 91–116, 2007.

[4] V. Giurgiutiu, "Tuned lamb wave excitation and detection with piezoelectric wafer active sensors for structural health monitoring," *Journal of Intelligent Material Systems and Structures*, vol. 16, no. 4, pp. 291–305, 2005.

[5] M. Mitra and S. Gopalakrishnan, "Guided wave based structural health monitoring: A review," *Smart Materials and Structures*, vol. 25, no. 5, p. 053001, 2016.

[6] C. Liu, J. B. Harley, M. Berges, D. W. Greve, and I. J. Oppenheim, "Robust ultrasonic damage detection under complex environmental conditions using singular value decomposition," *Ultrasonics*, vol. 58, pp. 75–86, Apr. 2015.

[7] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. ACM International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 245–250.

[8] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 32–43, 2014.

[9] Z. Su, L. Ye, and Y. Lu, "Guided lamb waves for identification of damage in composite structures: A review," *Journal of Sound and Vibration*, vol. 295, no. 3, pp. 753–780, 2006.

[10] C. G. Drury, P. Prabhu, and A. Gramopadhye, "Task analysis of aircraft inspection activities: methods and findings," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 34, no. 16. SAGE Publications, 1990, pp. 1181–1185.

[11] A. Riley, B. Aardema, P. Vosbury, M. Eiff, H. Frautschy, R. Serkenburg, D. Shaffer, T. Wild, and T. Michmerhuizen, *Aviation Maintenance Technician Handbook*, 2008.

[12] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[13] W. Shapiro, "The plane was disintegrating," *Time Magazine*, vol. 131, no. 19, 1988.

[14] J. Schmaltz, "Years after bridge collapse, questions and pain still linger," *New York Times*, 1984.

[15] C. R. Farrar and K. Worden, "An introduction to structural health monitoring," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, 2007.

[16] K. Worden, G. Manson, and D. Allman, "Experimental validation of a structural health monitoring methodology: Part i. novelty detection on a laboratory structure," *Journal of Sound and Vibration*, vol. 259, no. 2, pp. 323–343, 2003.

[17] Z. Su and L. Ye, *Identification of Damage using Lamb Waves: From Fundamentals to Applications*. Springer Science & Business Media, 2009, vol. 48.

[18] Y. Lu and J. E. Michaels, "A methodology for structural health monitoring with diffuse ultrasonic waves in the presence of temperature variations," *Ultrasonics*, vol. 43, no. 9, pp. 717–731, 2005.

[19] J. B. Harley, "Data-driven, sparsity-based matched field processing for structural health monitoring," Ph.D. dissertation, Carnegie Mellon University, 2014.

[20] D. Worlton, "Experimental confirmation of lamb waves at megacycle frequencies," *Journal of Applied Physics*, vol. 32, no. 6, pp. 967–971, 1961.

[21] A. Raghavan, "Guided-wave structural health monitoring," Ph.D. dissertation, The University of Michigan, 2007.

[22] T. Clarke, P. Cawley, P. D. Wilcox, and A. J. Croxford, "Evaluation of the damage detection capability of a sparse-array guided-wave shm system applied to a complex structure under varying thermal conditions," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 12, pp. 2666–2678, 2009.

[23] G. Konstantinidis, P. D. Wilcox, and B. W. Drinkwater, "An investigation into the temperature stability of a guided wave structural health monitoring system using permanently attached sensors," *IEEE Sensors Journal*, vol. 7, no. 5, pp. 905–912, 2007.

[24] S. J. Lee, N. Gandhi, J. S. Hall, J. E. Michaels, B. Xu, T. E. Michaels, and M. Ruzzene, "Baseline-free guided wave imaging via adaptive source removal," *Structural Health Monitoring*, vol. 11, no. 4, pp. 472–481, 2012.

[25] A. Croxford, P. Wilcox, B. Drinkwater, and G. Konstantinidis, "Strategies for guided-wave structural health monitoring," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2087. The Royal Society, 2007, pp. 2961–2981.

[26] A. J. Croxford, J. Moll, P. D. Wilcox, and J. E. Michaels, "Efficient temperature compensation strategies for guided wave structural health monitoring," *Ultrasonics*, vol. 50, no. 4, pp. 517–528, 2010.

[27] J. E. Michaels and T. E. Michaels, "Detection of structural damage from the local temporal coherence of diffuse ultrasonic signals," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 10, pp. 1769–1782, 2005.

[28] T. Clarke, F. Simonetti, and P. Cawley, "Guided wave health monitoring of complex structures by sparse array systems: influence of temperature changes on performance," *Journal of Sound and Vibration*, vol. 329, no. 12, pp. 2306–2322, 2010.

[29] J. B. Harley, C. Liu, I. J. Oppenheim, D. W. Greve, and J. M. Moura, "Coherent, data-driven lamb wave localization under environmental variations," *Proceedings of AIP Conference*, vol. 1650, no. 1, pp. 202–210, 2015. [Online]. Available: http://aip.scitation.org/doi/abs/10.1063/1.4914611

[30] C. Liu, J. B. Harley, N. O'Donoughue, Y. Ying, M. Bergés, M. H. Altschul, J. H. Garrett Jr, D. Greve, J. M. F. Moura, I. J. Oppenheim *et al.*, "Ultrasonic scatterer detection in a pipe under operating conditions using singular value decomposition," in *Review of Progress in Quantitative Nondestructive Evaluation: Vol 32*, vol. 1511, no. 1. AIP Publishing, 2013, pp. 1454–1461.

[31] M. Hilbert and P. López, "The worlds technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[32] K. Sayood, *Introduction to Data Compression*. Elsevier, 2012.

[33] M. A. Carreira-Perpinán, "A review of dimension reduction techniques," *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, vol. 9, pp. 1–69, 1997.

[34] I. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Laboratory, Tech. Rep., 2002.

[35] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[36] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, 2nd ed. John Wiley & Sons, 2013.

[37] J. A. Lee and M. Verleysen, "Unsupervised dimensionality reduction: overview and recent advances," in *Proceedings of Joint Conference on Neural Networks*. IEEE, 2010, pp. 1–8.

[38] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis (probability and mathematical statistics)*, 1st ed. Academic Press, 1979.

[39] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[40] W. B. Johnson and A. Naor, "The johnson-lindenstrauss lemma almost characterizes hilbert space, but not quite," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 885–891.

[41] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast johnson-lindenstrauss transform," in *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*. ACM, 2006, pp. 557–563.

[42] L. J. Schulman, "Clustering for edge-cost minimization," in *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*. ACM, 2000, pp. 547–555.

[43] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[44] T. Figiel, J. Lindenstrauss, and V. D. Milman, "The dimension of almost spherical sections of convex bodies," *Acta Mathematica*, vol. 139, no. 1, pp. 53–94, 1977.

[45] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," in *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 1. IEEE, 1998, pp. 413–418.

[46] K. Kang and G. Hooker, "Improving the recovery of principal components with semi-deterministic random projections," in *2016 Annual Conference on Information Science and Systems (CISS)*, March 2016, pp. 596–601.

[47] R. Hecht-Nielsen, "Context vectors: general purpose approximate meaning representations self-organized from raw data," *Computational Intelligence: Imitating Life*, pp. 43–56, 1994.

[48] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *European Conference on Computer Vision*. Springer, 2012, pp. 864–877.

[49] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, 2001, pp. 274–281.

[50] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of johnson and lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, 2003.

[51] A. Boisbunon, "The class of multivariate spherically symmetric distributions," http://aurelie.boisbunon.free.fr/downloads/loisSS.pdf, accessed: 2/14/2017.

[52] J. Matoušek, "On variants of the johnson–lindenstrauss lemma," *Random Structures & Algorithms*, vol. 33, no. 2, pp. 142–156, 2008.

[53] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[54] J. Hopcroft and R. Kannan, "Foundations of data science," 2014.

[55] L. CHEN, "Johnson-lindenstrauss transformation and random projection," http://www.math.uci.edu/ chenlong/MathPKU/JL.pdf, 2015, accessed: 2/10/2017.

[56] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 287–296.

[57] R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," in *Symposium on Foundations of Computer Science*. IEEE, 1999, pp. 616–623.